

AMENDMENTS TO THE CLAIMS

1. (original) A solid-state storage device comprising:

a physical electronic memory including a spare table region containing spare tables, a spare page region containing spare pages, and a data page region containing data pages;

an electronic memory interface that provides, to devices that access the electronic memory, memory operations directed to target data blocks specified by the accessing device via a logical data block address; and

a logic component that maps a logical data block address to a physical address describing the location of a data block in the electronic memory.

A2
2. (original) The solid-state storage device of claim 1 wherein a logical data block address comprises:

a page index that indexes a data page within the data page region, a spare table within the spare table region, and a spare page within the spare page region; and

a data block index that indexes a data block within the data page indexed by the page index and a spare table element within the spare table indexed by the page index.

3. (currently amended) The solid-state storage device of claim 2 wherein, in order to map a logical data block address to a physical address, the logic component:

extracts the page index and data block index from the logical data block address;

uses the page index to locate a corresponding spare table;

uses the data block index to locate a corresponding spare table element within the corresponding spare table;

when a status indication within the corresponding spare table element indicates that the logical data block address has been remapped,

uses a page offset within the corresponding spare table element and the page index extracted from the logical data block address to determine the physical

address of a spare data block, within a spare page, that corresponds to the logical data block address; and

when a status indication within the corresponding spare table element indicates that the logical data block address has not been remapped,

uses the page index and data block index to determine the physical address of a data block, within a data page, that corresponds to the logical data block address.

4. (original) The solid-state storage device of claim 2 wherein the electronic memory includes:

- A2
- a first spare table region;
 - a first spare page region;
 - a data page region;
 - a second spare page region; and
 - a second spare table region.

5. (original) The solid-state storage device of claim 4 wherein the first spare table region occupies a first, lower addressed portion of the electronic memory, the first spare page region occupies a second, next lower addressed portion of the electronic memory, the data page region occupies a third, middle portion of the electronic memory, the second spare page region occupies a fourth, higher addressed portion of the electronic memory, and the second spare table region occupies a fifth, highest addressed portion of the electronic memory.

6. (original) The solid-state storage device of claim 4 wherein redundant copies of the spare tables of the first spare table region are stored in the second spare table region.

7. (original) The solid state storage device of claim 4 further including:

- a spare table cache and a cached spare table identifier register.

8. (currently amended) The solid state storage device of claim 7 wherein, in order to map a logical data block address to a physical address, the logic component:

extracts the page index and data block index from the logical data block address;

uses the page index to locate a corresponding spare table;

determines, by comparing the page index extracted from logical data block address to the contents of the cached spare table identifier register, whether the spare table cache contains the contents of the located corresponding spare table;

when the spare table cache does not contain the contents of the located corresponding spare table,

~~copies~~ copies the located corresponding spare table into the spare table cache;

uses the data block index to locate a corresponding spare table element within the spare table cache;

when a status indication within the corresponding spare table element indicates that the logical data block address has been remapped,

uses a page offset within the corresponding spare table element and the page index extracted from the logical data block address to determine the physical address of a data block, within a spare page, that corresponds to the logical data block address; and

when a status indication within the corresponding spare table element indicates that the logical data block address has not been remapped,

uses the page index and data block index to determine the physical address of a data block, within a data page, that corresponds to the logical data block address.

9. (original) The solid-state storage device of claim 5 wherein a logical data block contains a 16-bit page index and a 16-bit data block index, wherein a data page contains up to 2^{16} data blocks, wherein a spare page contains up to $2^7 - 1$ data blocks, wherein the data page region contains up to 2^{16} data pages, wherein both the first and second spare page regions contain up to 2^{16} spare tables, wherein a spare table includes up to 2^{16} spare

table elements, and wherein both the first and the second spare table regions contain 2^{16} spare tables.

10. (currently amended) The solid-state storage device of claim 1 wherein a spare table contains a number of elements equal to the number of data blocks in a data page, wherein the spare table region contains a number of spare tables equal to the number of data pages within the data page region, wherein the spare page region contains a number of spare pages equal to the number of data pages within the data page region, and wherein a spare page contains a fixed number of data blocks, including a first data block that contains a spare block status map that contains spare block status map elements that store status information for data blocks stored within the spare page.

A2
11. (original) The solid-state storage device of claim 1 wherein a logical data block contains a 16-bit page index and a 16-bit data block index, wherein a data page contains up to 2^{16} data blocks, wherein a spare page contains up to $2^7 - 1$ data blocks, wherein the data page region contains up to 2^{16} data pages, wherein a spare page region contains up to 2^{16} spare tables, and wherein a spare table includes up to 2^{16} spare table elements.

12. (original) A method for transforming a logical data block address into a physical electronic memory address, the method comprising:

providing an electronic memory having a spare table region containing spare tables, a spare page region containing spare pages, and a data page region containing data pages;

extracting a page index and a data block index from the logical data block address; and

using the extracted page index and data block index to identify the address of a physical data block within the electronic memory corresponding to the logical data block address.

13. (original) The method of claim 12 further including:

using the extracted page index as an offset to locate a spare table within the spare table region;

using the extracted data block index as an offset to locate a spare table element within the spare table;

when a status indication within the spare table element indicates that the logical data block address has been remapped,

using the page index extracted from the logical data block address to determine the physical address of a spare page within the spare page region;

using a page offset within the spare table element to determine the physical address of a data block within the spare page; and

when a status indication within the corresponding spare table element indicates that the logical data block address has not been remapped,

using the page index from the logical data block address to determine the physical address of a data page within the data page region, and

using the data block index as an offset to determine the physical address of a data block within the data page.

14. (currently amended) The method of claim 13 further including:

when a status indication within the spare table element indicates that the logical data block address has been remapped,

using the status indication to determine whether the data block is in a low spare page region or a high spare page region and to select the indicated spare page region;

using the page index extracted from the logical data block address to determine the physical address of a spare page within the selected spare page region;

using a page offset within the spare table element to determine the physical address of a spare data block within the spare page.

15. (original) The method of claim 13 further including:

after using the extracted page index as an offset to locate a spare table within the spare table region, and before using the extracted data block index as an offset to locate a spare table element within the spare table,

checking a register to determine whether a currently cached spare table corresponds to the logical block address, and when the currently cached spare table does not correspond to the logical block address, copying the located spare table within the spare table region to a memory cache.

16. (original) The method of claim 15 further including:

when the currently cached spare table does not correspond to the logical block address, and when the contents of the currently cached spare table has been changed since the currently cached spare table was copied to the memory cache, copying the currently cached spare table back to the spare table region prior to copying the located spare table within the spare table region to the memory cache.

17. (original) A method for remapping a logical data block address to a different physical electronic memory address, the method comprising:

providing an electronic memory having a spare table region containing spare tables, a spare page region containing spare pages, and a data page region containing data pages;

extracting a page index and a data block index from the logical data block address;

using the extracted page index and data block index to select the address of a physical data block within a spare page in the spare page region of the electronic memory to which to remap the logical data block address; and

storing indications within an element of a spare table corresponding to the logical data block address that remap the logical data block address to the selected address.

18. (currently amended) The method of claim 17 further including:

using the extracted page index as an offset to locate a spare table within the spare table region;

using the extracted data block index as an offset to locate a spare table element within the spare table;

when a status indication within the spare table element indicates that the logical data block address has been remapped,

using the page index extracted from the logical data block address to determine the physical address of a spare page within the spare page region;

using a page offset within the spare table element to determine the physical address of a spare block status map element within a spare block status map in the first data block within the spare page corresponding to the logical data block address;

setting an indication in the spare block status map element to indicate that a corresponding data block within the spare page is no longer used;

searching the spare block status map within the spare block status map in the first data block within the spare page to select a spare block status map element indicating that a corresponding data block is available;

setting an indication in the selected spare block status map element indicating that the corresponding data block is in use; and

setting indications in the spare table element to indicate that the logical data block address is remapped to data block within the spare page corresponding to the selected spare block status map.
